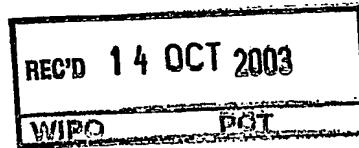


BUNDESREPUBLIK DEUTSCHLAND#1
#2

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen: 102 42 916.2 ✓

Anmeldetag: 16. September 2002

Anmelder/Inhaber: Siemens Aktiengesellschaft, München/DE

Bezeichnung: System zur Bereitstellung eines Standard-Frameworks für Automatisierungsgeräte

IPC: G 06 F 9/44

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 22. September 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

R. Baier

Brosig

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Beschreibung

System zur Bereitstellung eines Standard-Frameworks für Automatisierungsgeräte

5 Die Erfindung betrifft ein System sowie ein Verfahren zur Bereitstellung standardisierter, geräteunabhängiger Funktionalität für die Erstellung von Steuerungssoftware für Automatisierungssysteme.

10 In der heutigen Automatisierungswelt besitzen Automatisierungsgeräte unterschiedlichen Bautyps jeweils eigene Programmierschnittstellen und haben einen eigenen Befehlssatz, der bei der Programmierung bzw. Implementierung einer Automatisierungslösung auf dem jeweiligen Gerät verwendet wird.

15 Aufgrund dieser Unterschiede werden die im Rahmen einer Automatisierungslösung benötigten verschiedenen Automatisierungssysteme (Geräte einheitlichen Bautyps) unterschiedlich programmiert. Die Programmierung erfolgt dabei auf einem sehr niedrigen Niveau, das heißt es wird, vergleichbar zur Assemblerprogrammierung, eine einfache Syntax verwendet. Zur Realisierung einer komplexen Automatisierungslösung wird entsprechend eine Großzahl einfacher Befehle benötigt. Die so entstehenden Automatisierungslösungen sind spezifisch für die jeweiligen Automatisierungssysteme entwickelt und lassen sich, durch ihre Spezifität bedingt, nicht auf andere Automatisierungssysteme übertragen, welche jeweils wieder einen anderen Befehlssatz haben. Heutzutage werden also spezifische Lösungen mit hohem Programmieraufwand realisiert. Eine Übertragung von Programmen zwischen unterschiedlichen Automatisierungssystemen ist vor allem aufgrund der Tatsache nicht möglich, dass Basismechanismen der Automatisierung, wie das Ansprechen von Zählern oder das Herunterladen von Bausteinen auf jedem Gerätetyp unterschiedlich repräsentiert sind. Entsprechend werden die verschiedenen Automatisierungssysteme heutzutage als Einzelsysteme betrachtet. Jedes System hat seine eigene, spezielle Implementierung und eine eigene Pro-

grammierschnittstelle. Hieraus resultiert eine Vielzahl von spezifischen Compilern, die die jeweiligen Automatisierungsprogramme für entsprechende Zielsysteme übersetzen. Hinzu kommt, dass sich auch die Programmierumgebungen für die Erstellung einer Automatisierungslösung zwischen den einzelnen Automatisierungssystemen stark unterscheiden. So ist zum Beispiel das Download, also das Herunterladen einer erstellten Software, und das Debuggen, das Beheben von Fehlern im Programmcode für die jeweiligen Automatisierungssysteme unterschiedlich realisiert. Aus diesem Grund existieren Experten für die Programmierung einzelner Systeme. Nur wenige dieser Experten beherrschen verschiedene Automatisierungssysteme gleich gut und kennen deren jeweilige Spezifika.

- 15 Die Aufgabe der vorliegenden Erfindung ist es, eine standariserte und vereinfachte Programmierung und Implementierung von Automatisierungslösungen für verschiedene Automatisierungssysteme zu ermöglichen.
- 20 Diese Aufgabe wird gelöst durch ein System zur Bereitstellung von geräteunabhängiger Funktionalität für Automatisierungsgeräte mit ersten Mitteln zur Kapselung spezifischer Funktionen mindestens eines Automatisierungsgeräts und zum Bereitstellen von Basisfunktionalität des Automatisierungsgeräts und auf den ersten Mitteln aufsetzende zweite Mittel zur Bereitstellung von allgemeiner Funktionalität und/oder Automatisierungsfunktionalität.

Der Erfindung liegt die Erkenntnis zugrunde, dass derzeit im Automatisierungsumfeld in der Regel eine heterogene Hardwarewelt anzutreffen ist. Bei der Erstellung einer Automatisierungslösung für eine Anlage ist der Programmierer somit mit verschiedenen Automatisierungsgeräten konfrontiert, die jeweils eine eigene Sprache verstehen. Um die Erstellung einer Lösung zu vereinfachen, wird vorgeschlagen, eine einheitliche Umgebung zu schaffen, auf der Automatisierungslösungen für unterschiedliche Automatisierungssysteme auf standardisierte

Weise erstellt werden können. Hierzu wird ein so genanntes Standard-Framework für Automatisierungsgeräte bereitgestellt. Das Standard-Framework bildet eine Abstraktion der untergelegten Hardware, also der Automatisierungsgeräte und stellt hierbei gleichermaßen allgemeine Funktionalität, wie beispielsweise Download oder Ansprechen von Ein- und Ausgängen zur Verfügung, aber auch Automatisierungsfunktionalität, wie beispielsweise die Bereitstellung von Diagnosedaten oder das Forcen von Variablen. Das definierte Standard-Framework ist speziell auf die Belange der Automatisierungstechnik zugeschnitten und bietet unter anderem eine Vereinheitlichung der Runtime-Umgebung für verschiedene Automatisierungsgeräte sowie die Möglichkeit der Kapselung spezifischer Hardwareeigenschaften. Für die Erstellung des Programms, das auf einer speziellen Hardware läuft, werden hierbei Klassen des Frameworks genutzt. Eine Automatisierungslösung kann somit basierend auf den Funktionen und Automatisierungsobjekten des Standard-Frameworks realisiert werden. Die standardisierte Funktionalität des Frameworks wird hierbei von den Automatisierungsgeräten selbst angeboten. Hierzu sind auf den Automatisierungsgeräten Mittel zur Kapselung spezifischer Funktionen der Automatisierungsgeräte und zum Bereitstellen von Basisfunktionalität in Form des so genannten "Automation-Runtime-Layer" (ARL) realisiert. Der ARL bietet von der jeweiligen Hardware, also dem Automatisierungsgerät, abstrahierende, unabhängige Basisdienste an. Hierzu zählen beispielsweise Framing, Synchronisation, Speichermanagement, Merker sowie Zähler. Der ARL ist plattformspezifisch ausgebildet und besitzt die Aufgabe, alle plattformspezifischen Funktionen zu überdecken.

Auf dem ARL setzen Mittel zur Bereitstellung allgemeiner Funktionalität in Form des so genannten "Runtime-Framework" (RF) auf. Das RF bietet hierbei höhere Dienste an und stellt neben allgemeiner Funktionalität (wie Speicherverwaltung, Threading, Synchronisation, ...) auch Automatisierungsfunktionalität zur Verfügung. Im Gegensatz zum ARL ist die Implementierung des RF nicht plattformspezifisch. Hierdurch kann eine einmalige Implementierung mit geringem Aufwand auf ver-

schiedene Zielplattformen, das heißt Automatisierungssysteme, portiert werden. Vorteilhaft ist hierbei vor allem, dass bei der Erstellung von Automatisierungslösungen eine Bereitstellung von allgemeinen Automatisierungsdiensten, die generell verwendet werden können, ermöglicht wird. Die Nutzung des vorgestellten Systems führt zu einer Vereinfachung der Implementierung einzelner Automatisierungskomponenten. Es erfolgt eine Standardisierung der Services zum Zugriff auf die Automatisierungsgeräte und zum Zugriff auf automatisierungsspezifische Objekte, Alarme und Prozesswerte. Durch die Abstraktion von der spezifischen Hardware, das heißt eine Unabhängigkeit von speziellen Automatisierungssystemen, wird eine hardwareunabhängige Programmierung der Automatisierungskomponenten ermöglicht. Spezifisches, auf ein Automatisierungssystem zugeschnittenes Expertenwissen, ist für die Erstellung der Automatisierungslösung hierdurch nicht mehr nötig. Ein aufwendiges Erstellen von speziellem Programmcode, der auf relativ niedrigem Niveau ist, erübrigt sich durch die Nutzung des vorgestellten Systems, es ermöglicht vielmehr eine Programmierung unter Nutzung von Hochsprachenprogrammiersprachen.

Dadurch wird dem Programmierer ermöglicht, sich auf die Lösung der eigentlichen automatisierungsspezifischen Probleme zu konzentrieren, statt seine Kapazität in die Erzeugung von rudimentären Programmbestandteilen investieren zu müssen. Des Weiteren sorgt das Standard-Framework für eine Durchgängigkeit bei der Diagnose und den Fehleranalysen über die Grenzen der Automatisierungssysteme hinweg. Auch das Zusammenspiel verschiedener Komponenten, die in einer Automatisierungslösung realisiert sind, ist einfacher zu analysieren bzw. zu debuggen. Durch die Nutzung des Standard-Frameworks bietet sich die Möglichkeit, höherwertige Zusatzpakete zu entwickeln, die Objekte oder Funktionen anbieten, welche dann ebenfalls unabhängig von der Zielplattform sein können. Solche technologischen Objekte oder Zusatzpakete können innerhalb des Systems in Form von Bibliotheken hinterlegt werden, um für wiederkehrende Automatisierungsaufgaben zur Verfügung zu stehen. Zudem ermöglicht die Verwendung des Standard-

Frameworks ein Portieren von Automatisierungslösungen über verschiedene Gerätefamilien hinweg.

Eine weitere vorteilhafte Ausbildung der Erfindung ist da-
5 durch gekennzeichnet, dass die ersten Mittel zur Überdeckung spezifischer Funktionen als automatisierungsgerätespezifische Adapter ausgebildet sind. Die automatisierungsgerätespezifischen Eigenschaften werden vom ARL dadurch auf relativ niedriger Ebene im Gesamtsystem standardisiert, so dass sie in
10 Form von, für ein Programmiersystem allgemein verständlichen, Befehlen zur Verfügung stehen. Hierdurch ist gewährleistet, dass alle Automatisierungsgeräte einheitlich auf niedriger Ebene angesprochen werden können. Eine Standardisierung auf höherer Hierarchieebene innerhalb eines Automatisierungssys-
15 tems würde unter Umständen zu einem Verlust von automatisie- rungsgerätespezifischen Eigenschaften führen. Die Verwendung von spezifischen Adapters sorgt hingegen dafür, dass alle Ei- genschaften auch in allgemeiner Form zur Verfügung gestellt werden können.
20 Eine weitere vorteilhafte Ausbildung des Systems ist dadurch gekennzeichnet, dass die Mittel zur Bereitstellung von allge- meiner Funktionalität und/oder Automatisierungsfunktionalität geräteunabhängig ausgebildet sind. Vorteilhaft ist hierbei, dass allgemeine Funktionalität wie beispielsweise die Realisierung eines Downloads nicht in dem gerätespezifischen ARL integriert werden muss. Besonders Automatisierungsfunktiona- lität, die unabhängig von spezifischen Automatisierungsgerä- ten ist, wie beispielsweise das Durchführen von Diagnosen,
30 muss nicht auf den spezifischen ARLs realisiert sein. Dadurch dass diese Funktionalitäten geräteunabhängig sind, müssen sie nur einmal implementiert werden und können dann automatisie- rungsgerätespezifisch unabhängig verwendet werden. Eine ein- malige Erstellung des Runtime-Framework (RF) wird hierdurch
35 ermöglicht.

Eine weitere vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass das System zur Nutzung durch ein Erstellssystem für die Erstellung von Steuerungssoftware vorgesehen ist. Die vom System bereitgestellte standardisierte 5 sowie geräteunabhängige Funktionalität kann dadurch vorteilhaft von einem gängigen automatisierungstechnischen Editor, einem Programmierwerkzeug oder allgemein, einem Engineering-System verwendet werden. Eine einfachere, schnellere und kostengünstigere Herstellung von Steuerungssoftware wird durch 10 diese vorteilhafte Ausbildung ermöglicht.

Eine weitere vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass das System zur Bereitstellung von technologischen Objekten für Automatisierungsgeräte vorgesehen ist. Objekte wie beispielsweise ein Hubtisch, ein Förderband oder Kessel können, wenn sie sich in Form von allgemein wieder verwendbaren Funktionen abbilden lassen in Form einer Bibliothek abgelegt werden. Diese technologischen Objekte stehen dem Programmierer dann standardmäßig mit ihren Eigenschaften und Methoden für die Realisierung einer Automatisierungslösung zur Verfügung. Der Programmieraufwand wird durch 20 diese vorgegebenen Objekte erheblich verringert.

Eine weitere vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass ein Speicher zur Speicherung von Automatisierungslösungen für wiederkehrende Aufgaben vorgesehen ist. Die bereits realisierten Automatisierungslösungen können auf dem Automatisierungsgerät hinterlegt werden, und stehen für die Wiederverwendung in neuen Automatisierungslösungen zur Verfügung. Einem zur Erstellung von Steuerungssoftware genutzten Engineering-System können sind sie bekannt. Eine schnellere Entwicklung und auch die Nutzung bereits vorhandenen Lösungswissens wird hierdurch ermöglicht. 30

35 Eine weitere vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass das System zur Nutzung eines Internet und/oder Intranet zur Übertragung von Daten vorgesehen

ist. Hierdurch wird beispielsweise eine vom Ort der Automatisierungsgeräte unabhängige Installation eines zur Programmierung verwendeten Engineering-Systems ermöglicht. Eine erstellte Automatisierungslösung kann über das Internet oder 5 Intranet an die jeweiligen Zielautomatisierungsgeräte gesendet werden. Die Versendung von programmierten Lösungen in Form von Disketten oder anderen Datenträgern erübrigts sich durch diese Ausbildung der Erfindung.

10 Eine weitere vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass eine automatisierungsspezifisch ausgebildete Programmiersprache für die Erstellung einer Steuerungssoftware als Zwischensprache für Automatisierungssprachen, als Zielsprache von Erstellsystemen und zur Abbildung auf die ersten Mittel (dem RF) auf einem Automatisierungsgerät als Zielplattform vorgesehen ist. Die Realisierung dieser automatisierungsspezifisch ausgebildeten Zwischensprache für Automatisierungslösungen bietet den Vorteil, dass 15 spezifische Programmiersprachen, die heutzutage in der Automatisierungstechnik von den Erstellsystemen verwendet werden, auf eine einzige Programmiersprache als Zielsprache abgebildet werden können. Vorteilhaft ist hierbei, dass für die Lösung einer Automatisierungsaufgabe für die Erstellung geeignete Editoren und Erstellsysteme verwendet werden können. Diese Editoren und Erstellsysteme haben jeweils ihre eigene 20 Programmiersprache, die dann in die Zwischensprache für Automatisierungssprachen (AIL = Automation Intermediate Language) übersetzt wird. Durch die anschließende Nutzung der AIL wird gewährleistet, dass die speziellen Automatisierungsaufgaben, 25 die mit Hilfe der jeweiligen Editoren gelöst wurden, später problemlos zusammenarbeiten. Vorteilhaft ist hierbei außerdem, dass es zu einer Abstraktion der Programmierwerkzeuge von der zugrundeliegenden Hardware kommt. Die Nutzung der AIL ermöglicht auch das einheitliche Debuggen bzw. die Fehleranalyse 30 der erstellten Automatisierungslösung, da die unterschiedlichen Programmiersprachen zuvor in eine einheitliche Sprache übersetzt worden sind. Die Nutzung der AIL ermöglicht 35

somit erstmals, einen Standard-Debugger zu verwenden. Hierdurch wird der Zugriff vom PC bis hinunter auf die Steuerung ermöglicht.

- 5 Eine weitere vorteilhafte Ausbildung des Systems ist dadurch gekennzeichnet, dass Compiler (5) zur Abbildung der Sprache auf die Zielplattform vorgesehen sind. Die Programmierwerkzeuge müssen sich ausschließlich auf die Abbildung in die AIL konzentrieren. Die Übersetzung in die hardwarespezifische
- 10 Version erfolgt durch die speziellen Compiler. Eine Realisierung der entsprechenden Lösung auf eine bestimmte Zielplattform hin muss nicht erfolgen. Hierdurch wird die Grundlage für die Erstellung hardwareunabhängiger Automatisierungskomponenten gebildet. Eine Automatisierungslösung läuft auf einem Automatisierungsgerät genau dann, wenn die Anforderungen an den jeweiligen ARL erfüllt sind. Die Anzahl der dafür benötigten Compiler verringert sich, da nicht jede mögliche Editor-Automatisierungssystem-Kombination ihren eigenen Compiler benötigt. Es werden nur System- bzw. Editorenspezifische Compiler in die AIL und von der AIL benötigt.
- 15
- 20

Im Folgenden wird die Erfindung anhand der in den Figuren dargestellten Ausführungsbeispiele näher beschrieben und erläutert. Es zeigen:

FIG 1 eine schematische Darstellung eines Ausführungsbeispiels des Standard-Frameworks für Automatisierungssysteme,

30 FIG 2 eine schematische Darstellung der Verwendung einer automatisierungsspezifischen Zwischensprache.

FIG 1 zeigt eine schematische Darstellung eines Ausführungsbeispiels eines Standard-Frameworks zur Bereitstellung geräteunabhängiger, standardisierter Funktionalität für unterschiedliche Automatisierungssysteme $6_{1..n}$. Auf den Automatisierungssystemen $6_{1..n}$ setzen systemspezifische Adapter $3_{1..n}$, so genannte Automation-Runtime-Layer (ARL) auf. Die ARL $3_{1..n}$

überdecken automatisierungsgerätespezifische Funktionen und stellen Basisfunktionalität der Automatisierungssysteme 6_{1..n} zur Verfügung. Auf den Adaptern 3_{1..n} setzen automatisierungssystemunabhängige Runtime-Frameworks (RF) 4 auf. Die RFs 4 stellen allgemeine Funktionalität und Automatisierungsfunktionalität zur Verfügung. Die Automatisierungslösung wird mit Hilfe eines Engineering-Systems 2 erstellt. Hierzu kann auf eine Bibliothek mit technologischen Objekten 7 zurückgegriffen werden. Bereits realisierte Automatisierungslösungen können in einem Speicher 8 auf den Automatisierungsgeräten zur Wiederverwendung hinterlegt werden. Die mittels des Engineering-Systems erstellte Software wird mit Hilfe der Compiler 5_{1..n} für die jeweiligen Automatisierungssysteme 6_{1..n} in eine Runtimeversion übersetzt. Die Daten werden mit Hilfe einer Datenübertragungsvorrichtung 9 übertragen.

Die Besonderheit des in FIG 1 dargestellten Systems 1 besteht darin, dass auf einem beliebigen Engineering-System 2 eine standardisierte, geräteunabhängige Automatisierungslösung erstellt werden kann. Diese Besonderheit wird dadurch gewährleistet, dass die automatisierungssystemspezifischen Eigenschaften mit Hilfe der ARL 3_{1..n} überdeckt werden. Im Zusammenwirken mit dem systemunabhängigen RF 4, welches weitere Funktionalität bereitstellt, steht auf den Automatisierungsgeräten 6_{1..n} eine vereinheitlichte Ebene zur Verfügung, auf der mit Hilfe höherwertiger Programmiersprachen eine Automatisierungslösung realisiert werden kann. Dies ermöglicht die Verwendung allgemeiner Bausteine, wie beispielsweise technologischen Objekten 7, die dem Engineering-System zur Verfügung stehen. Eine spezifische Implementierung der jeweiligen Lösung für die einzelnen Automatisierungssysteme 6_{1..n} erübrigt sich auf diese Weise. Einmal entwickelte Lösungen können geräte- und systemunabhängig wieder verwendet werden. Der jeweilige Compiler 5_{1..n} sorgt dann für eine entsprechende Übertragung des allgemeinen Programmcodes in eine vom jeweiligen Automatisierungssystem 6_{1..n} interpretierbare Form. Allgemein wird durch das vorgestellte System die Implementierung einer

Automatisierungslösung enorm vereinfacht: Automatisierungs-systemspezifisches Expertenwissen ist nicht mehr notwendig und eine Wiederverwendung bereits implementierter Lösungen ist möglich.

5 FIG 2 zeigt die beispielhafte Verwendung einer automatisierungsspezifischen Programmiersprache als Zwischensprache 10 zwischen Programmiersprachen automatisierungstechnischer Editoren 12 und automatisierungsgerätespezifischen Sprachen. Die
10 mittels automatisierungsspezifischer Editoren 12 erstellten Programmcodes werden mit Hilfe von spezifischen Compilern 11 in die Zwischensprache 10 übersetzt. Aus der Zwischensprache heraus werden mittels Compiler 5_{1..n} die Anweisungen in automatisierungssystemspezifische Anweisungen übersetzt. Die Daten
15 können hierbei auf die Automatisierungsgeräte 6_{1..n} über ein Intranet bzw. Internet 13 übertragen werden.

Der wesentliche Vorteil der in FIG 2 dargestellten schematischen Ausbildung des Systems mit einer Zwischensprache 10 für
20 Automatisierungssysteme (AIL = Automation Intermediate Language) besteht darin, dass die Menge der zu verwendenden Compiler, die für die Erstellung einer Automatisierungslösung verwendet werden müssen, wesentlich reduziert werden kann.
Ohne die Verwendung einer AIL muss jede verwendete Programmiersprache eines automatisierungsspezifischen Editoren 12_{1..m} mittels eines speziellen Compilers für die konkrete Zielplattform 6_{1..n} übersetzt werden. Hieraus ergibt sich eine Menge von n*m Compilern für eine Menge von m Editoren und n Automatisierungsgeräten. Bei der Verwendung der Zwischensprache 10 übersetzen die Compiler 11_{1..m} ihren Quellcode in die Zwischensprache 10. Von der Zwischensprache wird der Code mit Hilfe systemspezifischer Compiler 5_{1..n} in die Sprache der Zielplattform 6_{1..n} übersetzt. Bei der Verwendung der AIL sind somit nur noch n+m Compiler notwendig, um eine spezifische Automatisierungslösung zu realisieren. Außerdem kann ein Großteil der Realisierung innerhalb der AIL selbst erstellt
30
35

werden. Somit vereinfacht sich die Erstellung einer Automatisierungslösung um ein Vielfaches.

Zusammenfassend betrifft die Erfindung ein System sowie ein
5 Verfahren zur Bereitstellung von standardisierter, geräteun-
abhängiger Funktionalität für das Erstellen einer Lösung im
Automatisierungsumfeld. Hierbei werden die spezifischen Ei-
genschaften und Funktionen unterschiedlicher Automatisie-
rungssysteme $6_{1..n}$ auf den jeweiligen Geräten direkt durch ei-
10 nen systemspezifischen Adapter, dem Automation-Runtime-Layer
(ARL) $3_{1..n}$ abgedeckt, bzw. generalisiert. Eine weitere, auf
dem jeweiligen Gerät implementierte Komponente, das Runtime-
Framework (RF) 4 stellt allgemeine Automatisierungs- und/oder
Basisfunktionalität zur Verfügung. Die Automatisierungslösung
15 kann entsprechend standardisiert auf einem Engineering-System
2 erstellt unabhängig von der jeweiligen Zielplattform $6_{1..n}$
beliebig portiert werden. Die Nutzung einheitlicher Objekt-
bibliotheken 7 und die Wiederverwendung einmal erstellter Lö-
sungen ist systemunabhängig möglich.

Patentansprüche

1. System zur Bereitstellung von geräteunabhängiger Funktionalität für Automatisierungsgeräte mit
 - 5 - ersten Mitteln (3) zur Kapselung spezifischer Funktionen mindestens eines Automatisierungsgeräts (6) und zum Bereitstellen von Basisfunktionalität des Automatisierungsgeräts (6) und
 - auf den ersten Mitteln (3) aufsetzende zweite Mittel (4) zur Bereitstellung von allgemeiner Funktionalität und/oder Automatisierungsfunktionalität.
2. System nach Anspruch 1,
durch gekennzeichnet,
15 dass die ersten Mittel (3) als automatisierungsgerätespezifische Adapter ausgebildet sind.
3. System nach einem der vorhergehenden Ansprüche,
durch gekennzeichnet,
20 dass die zweiten Mittel (4) geräteunabhängig ausgebildet sind.
4. System nach einem der vorhergehenden Ansprüche,
durch gekennzeichnet,
dass das System zur Nutzung durch ein Erstellsystem (2) für
die Erstellung von Steuerungssoftware vorgesehen ist.
5. System nach einem der vorhergehenden Ansprüche,
durch gekennzeichnet,
30 dass das System zur Bereitstellung von technologischen Objekten (7) für Automatisierungsgeräte (6_{1..n}) vorgesehen ist.
6. System nach einem der vorhergehenden Ansprüche,
durch gekennzeichnet,
35 dass ein Speicher (8) zur Speicherung von Automatisierungslösungen für wiederkehrende Aufgaben vorgesehen ist.

7. System nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
dass das System zur Nutzung eines Internet und/oder Intranet
(13) zur Übertragung von Daten vorgesehen ist.

5 8. Verfahren zur Bereitstellung von geräteunabhängiger Funktionalität für Automatisierungsgeräte bei dem

- spezifische Funktionen mindestens eines Automatisierungsgeräts (6) überdeckt werden und Basisfunktionalität des Automatisierungsgeräts (6) bereitgestellt wird,
- allgemeine Funktionalität und/oder Automatisierungsfunktionalität bereitgestellt wird.

15 9. Verfahren nach Anspruch 8,
dadurch gekennzeichnet,
dass automatisierungsgerätespezifische Adapter auf den Automatisierungsgeräten (6) verwendet werden.

20 10. Verfahren nach einem der Ansprüche 8 oder 9,
dadurch gekennzeichnet,
dass die allgemeine Funktionalität und/oder Automatisierungsfunktionalität geräteunabhängig bereitgestellt wird.

) 11. Verfahren nach einem der Ansprüche 8 bis 10,
dadurch gekennzeichnet,
dass ein Erstellsystem (2) für die Erstellung von Steuerungssoftware genutzt wird.

30 12. Verfahren nach einem der Ansprüche 8 bis 11,
dadurch gekennzeichnet,
dass technologische Objekte (7) für Automatisierungsgeräte (6_{1..n}) bereitgestellt werden.

35 13. Verfahren nach einem der Ansprüche 8 bis 12,
dadurch gekennzeichnet,
dass Automatisierungslösungen für wiederkehrende Aufgaben gespeichert werden.

14. Verfahren nach einem der Ansprüche 8 bis 13, dadurch gekennzeichnet, dass eine Internet und/oder Intranet (13) zur Übertragung von Daten genutzt wird.

5

15. Automatisierungsspezifisch ausgebildete Programmiersprache für die Erstellung einer Steuerungssoftware für ein System sowie für ein Verfahren nach den Ansprüchen 1 bis 14.

10 16. Programmiersprache nach Anspruch 15,

dadurch gekennzeichnet,
dass die Programmiersprache

- als Zwischensprache für Automatisierungssprachen,
- als Zielsprache von Erstellsystemen (2) und

15 - zur Abbildung auf die ersten Mittel (3) auf einem Automatisierungsgerät (6) als Zielplattform vorgesehen ist.

17. Programmiersprache nach den Ansprüchen 15 oder 16,

dadurch gekennzeichnet,

20 dass Compiler (5) zur Abbildung der Programmiersprache auf die Zielplattform vorgesehen sind.

18. Automatisierungsgerät mit

- ersten Mitteln (3) zur Überdeckung spezifischer Funktionen und zum Bereitstellen von Basisfunktionalität des Automatisierungsgeräts (6) und
- auf den ersten Mitteln aufsetzende zweite Mittel (4) zur Bereitstellung von allgemeiner Funktionalität und/oder Automatisierungsfunktionalität.

30

19. Automatisierungsgerät nach Anspruch 18,

dadurch gekennzeichnet,

dass die ersten Mittel (3) als automatisierungsgerätespezifische Adapter ausgebildet sind.

35

20. Automatisierungsgerät nach Anspruch 18,

dadurch gekennzeichnet,

dass die zweiten Mittel (4) geräteunabhängig ausgebildet sind.

Zusammenfassung

System zur Bereitstellung eines Standard-Frameworks für Automatisierungsgeräte

5 Die Erfindung betrifft ein System sowie ein Verfahren zur Bereitstellung von standardisierter, geräteunabhängiger Funktionalität für das Erstellen einer Lösung im Automatisierungsumfeld. Hierbei werden die spezifischen Eigenschaften und
10 Funktionen unterschiedlicher Automatisierungssysteme ($6_{1..n}$) auf den jeweiligen Geräten direkt durch einen systemspezifischen Adapter, dem Automation-Runtime-Layer (ARL) ($3_{1..n}$) abgedeckt, bzw. generalisiert. Eine weitere, auf dem jeweiligen Gerät implementierte Komponente, das Runtime-Framework (RF)
15 (4) stellt allgemeine Automatisierungs- und/oder Basisfunktionalität zur Verfügung. Die Automatisierungslösung kann entsprechend standardisiert auf einem Engineering-System (2) erstellt unabhängig von der jeweiligen Zielplattform $6_{1..n}$ beliebig portiert werden. Die Nutzung einheitlicher Objektbibliotheken (7) und die Wiederverwendung einmal erstellter Lösungen ist systemunabhängig möglich.
20

Fig. 1

200211020

1/1



FIG 1

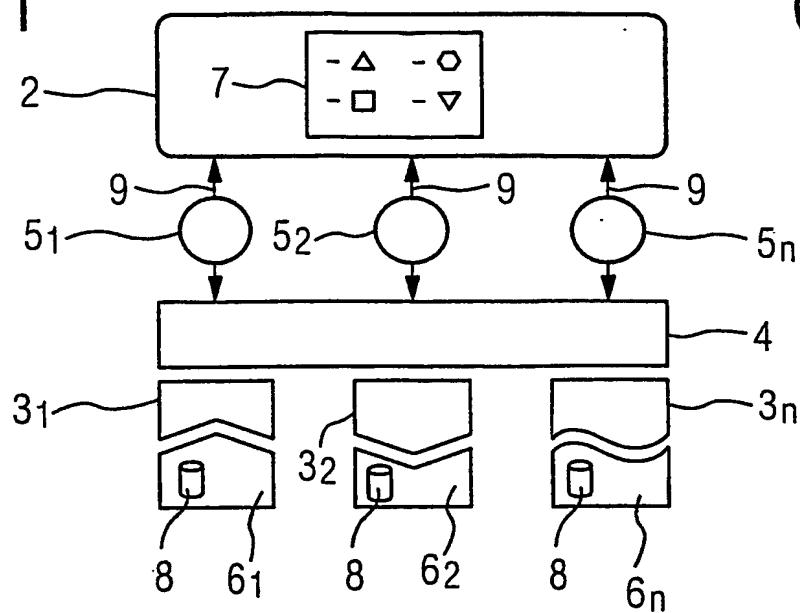
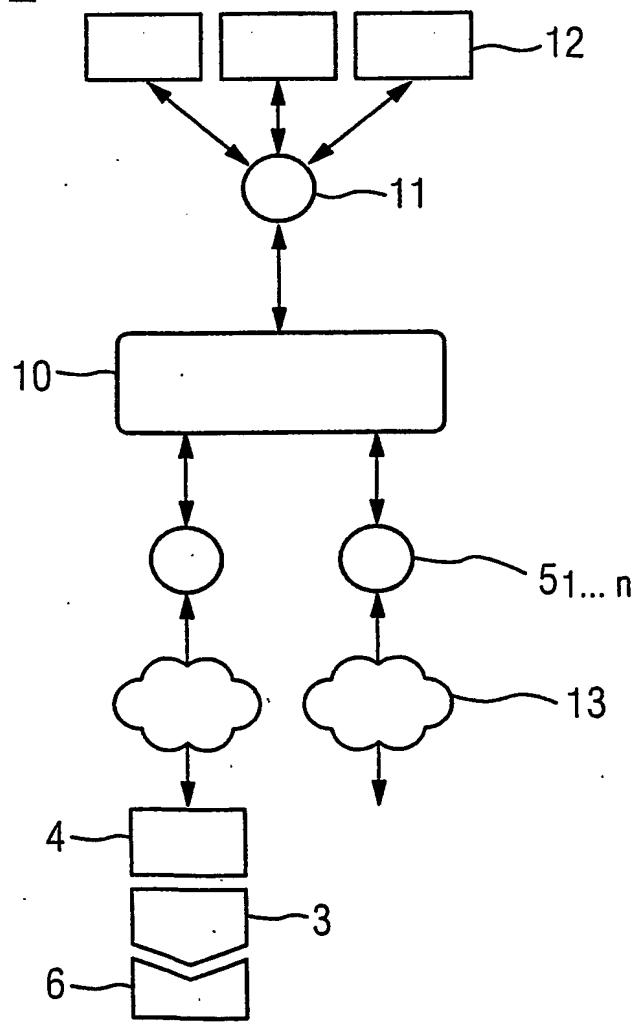


FIG 2



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: hole-punched overfeet**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.